

Chapter 1

Incorporation of Region of Interest in a Decomposition-Based Multi-Objective Evolutionary Algorithm

Ivan Reinaldo Meneghini and Frederico Gadelha Guimarães and António Gaspar-Cunha and Miri Weiss Cohen

Abstract Preference-based Multi-Objective Evolutionary Algorithm (MOEA) restrict the search to a given region of the Pareto front preferred by the Decision Maker (DM), called the Region of Interest (ROI). In this paper, a new preference-guided MOEA is proposed. In this method, we define the ROI as a preference cone in the objective space. The preferential direction and the aperture of the cone are parameters that the DM has to provide to define the ROI. Given the preference cone, we employ a weight vector generation method that is based on a steady-state evolutionary algorithm. The main idea of our method is to evolve a population of weight vectors towards the characteristics that are desirable for a set of weight vectors in a decomposition-based MOEA framework. The main advantage is that the DM can define the number of weight vectors and thus can control the population size. Once the ROI is defined and the set of weight vectors are generated within the preference cone, we start a decomposition-based MOEA using the provided set of weights in its initialization. Therefore, this enforces the algorithm to converge to the ROI. The results show the benefit and adequacy of the preference cone MOEA/D for preference-guided many-objective optimization.

Ivan Reinaldo Meneghini
Instituto Federal de Educação Ciência e Tecnologia de Minas Gerais (IFMG) Campus Ibirité, 2
Mato Grosso St. 32407-190, Ibirité, Minas Gerais, Brazil, e-mail: ivan.reinaldo@ifmg.edu.br

Frederico Gadelha Guimarães
Department of Electrical Engineering, Universidade Federal de Minas Gerais (UFMG),
6627 Pres. Antônio Carlos Ave. 31270-901, Belo Horizonte, Minas Gerais, Brazil, e-mail:
fredericoguimaraes@ufmg.br

António Gaspar-Cunha
Institute of Polymers and Composites, University of Minho, Campus de Azurém, 4800-058
Guimarães, Portugal e-mail: agc@dep.uminho.pt

Miri Weiss Cohen
Department of Software Engineering, Braude College of Engineering, Karmiel, Israel, Snunit St
51, Karmiel, 2161002, Israel e-mail: miri@braude.ac.il

1.1 Introduction

Multi-objective evolutionary algorithms (MOEA) are recognized as suitable methods to find high quality approximations to the set of solutions to multi-objective optimization problems [18]. These optimal solutions, known as Pareto optimal solutions, are characterized by the trade-off relation between the conflicting objectives, such that some improvement in one objective function must lead to deterioration in at least one of the other objectives. However, as the number of objectives grows, we reach the field of many-objective optimization problems (MaOPs) [10]. This boundary is usually defined when the number of objective functions is greater than three or four, given empirical studies about the downgrading performance of most multi-objective algorithms when the number of objectives increase, see for instance [2].

Without any prior preference provided by the decision-maker (DM), MOEA are designed to find an unbiased, well-distributed approximation of the entire Pareto Front (PF), a task that becomes increasingly harder in MaOPs. This brings a number of challenges related to converging to such a large set of solutions, visualizing solutions found, performing decision-making with a large number of alternatives [10]. Moreover, a high computational cost of properly sampling of the high-dimensional Pareto front. For this reason, many preference-based MOEA have been proposed in the literature [3], designed to converge to a subset of Pareto-optimal solutions located at a given region of the PF preferred by the DM, usually called Region of Interest (ROI). These preference-based MOEA are an intermediate approach for incorporating preferences in multi-objective optimization: *a priori* information is needed to define the ROI and, after some desirable solutions are found, the DM can select the most satisfying one *a posteriori* or restart the process by adjusting the ROI, hence following an *interactive* approach. With this novel approach, one can avoid the main disadvantages of the *a priori* methods. Defining the ROI might be easier for the DM than modelling the preferences into specific parameters of a parameterized single-objective optimization problem. Furthermore, it can alleviate the high computational cost and time consumption of full *a posteriori* methods.

The proposed methodology in this paper is therefore an intermediate approach for incorporating preferences into many-objective optimization problems. Its' a methodology aligned with the trend of interactive approaches, and follows the framework of any MOEA based on decomposition.

In the last decade, MOEA based on decomposition/aggregation methods have attracted the attention of the evolutionary multi-objective optimization community, with several studies to show their potential and limitations, and to improve their performance in constrained multi- and many-objective optimization problems [16]. The decomposition-based MOEA rely on aggregation functions that are based on different weight vectors. Those weight vectors might represent a weighted aggregation of objectives or a preference direction in objective space depending on the interpretation and the context of the decomposition method adopted within the algorithm. The key point is that the weight vector generation method is the primary step

in decomposition-based MOEA, affecting the diversity of the Pareto approximation and overall performance of the algorithm.

In this paper, we define the ROI as a preference cone in the objective space. The preference cone could be defined by a preferential direction vector \mathbf{v} , which corresponds to the axis of the cone, having the origin or utopian point as the apex, and the angle τ between the axis and the generating lines (generatrix). The preferential direction and the aperture of the cone are the parameters that the DM has to provide to define the ROI. Given the preference cone, we employ a weight vector generation method that is based on a steady-state evolutionary algorithm. The main idea of our method is to evolve a population of weight vectors towards the characteristics that are desirable for a set of weight vectors in a decomposition-based MOEA framework. Once the ROI is defined and the set of weight vectors are generated within the preference cone, we start a decomposition-based MOEA using the provided set of weights in its initialization. Therefore, this enforces the algorithm to converge to the preference cone, which in turn represents the ROI to the DM.

1.2 Background

1.2.1 Many-objective optimization

A multi-objective optimization problem (MOP) [18] is defined by:

$$\begin{aligned} \mathbf{x}^* &= \arg \min F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{subject to: } &\begin{cases} G(\mathbf{x}) \leq 0, \\ H(\mathbf{x}) = 0, \\ \mathbf{x} \in \Omega \end{cases} \end{aligned} \quad (1.1)$$

where $\mathbf{x} \in \Omega$ are the decision variables in the decision space Ω . Their image, $\mathbf{y} = F(\mathbf{x})$ given by the function F , is the objective space. The functions $G(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_P(\mathbf{x}))$ and $H(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_Q(\mathbf{x}))$ define the inequality and equality constraints respectively. The constraint functions define the feasible set $\Omega \subseteq X$ and the feasible region in the objective space $F(\Omega) \subseteq Y$. This paper will consider only the case where $X \subseteq \mathbb{R}^N$ and $Y \subseteq \mathbb{R}^M$. In the application $F : X \rightarrow Y$, each coordinate $f_i(\mathbf{x})$ of $F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))$ is an objective function of the MOP defined in (1.1).

The solution of problem (1.1) uses the following relationship between \mathbb{R}^M vectors: Let $\mathbf{u} = (u_1, \dots, u_M)$ and $\mathbf{v} = (v_1, \dots, v_M)$ be vectors in \mathbb{R}^M . Then $\mathbf{u} < \mathbf{v}$ if and only if $u_i \leq v_i \forall i \in \{1, \dots, M\}$ and $\exists i \in \{1, \dots, M\}$ such that $u_i < v_i$.

If $\mathbf{u} < \mathbf{v}$ then \mathbf{u} (Pareto) dominates \mathbf{v} and if neither $\mathbf{u} < \mathbf{v}$ nor $\mathbf{v} < \mathbf{u}$ then \mathbf{u} and \mathbf{v} are said to be non-dominated. In this case, the solution of the MOP (1.1) is defined by the set $\mathcal{P} = \{\mathbf{x} \in \Omega; \nexists \mathbf{s} \in \Omega \text{ such that } F(\mathbf{s}) < F(\mathbf{x})\}$. This set is called the Pareto-optimal Set of the MOP (1.1) and its elements are minimal solutions by the partial order $<$. The image $F(\mathcal{P})$ of the points in \mathcal{P} in the objective space Y is

(in general) an $(M - 1)$ -dimensional manifold (for continuous problems), called the Pareto Front (PF).

Obtaining the exact solution of a MOP is a very difficult task. Since the objectives in a MOP are conflicting and the order relation that establishes the best solution is a partial order. There is no single solution for a given MOP, but a set of non-dominated solutions of large or even infinite cardinality. The general desirable solution of a MOP consists of an approximation of a subset of \mathcal{P} (or $F(\mathcal{P})$) with the following important characteristics:

- the approximation set is sufficiently close to a subset of the Pareto Front;
- the approximation set presents maximum coverage of the Pareto Front.

This second item guarantees the existence of feasible solutions in any part of the Pareto Front. In a hypothetical situation of no preference on the part of the decision maker, any of these solutions can be chosen arbitrarily.

An efficient way of determining an approximation of the solution of these problems is through Multi-Objective Evolutionary Algorithms (MOEA) [18]. In this methodology, in each iteration a set of new candidate solutions is produced from the current population in order to determine, a set of non-dominated points (from the objective space sets). The main difference between the types of MOEA is according to the selection of a new population. These algorithms are categorized as follows[18]:

- Decomposition-based MOEA: In this approach, a set of weight vectors (or direction vectors) are created along the objective space and associated to the population. Then, the MOP is decomposed into a number of Single Objective Problems, each one representing a parameterized scalarizing function. MOEA/D [17] and NSGA-III [6] are examples of algorithms that use this method.
- Dominance-based MOEA: In this approach, all the objectives are optimized simultaneously and the new individuals in the populations are selected using the dominance relation. NSGA-II [5] and SPEA2 [20] are examples of algorithms that use this method.
- Indicator-based MOEA: In this approach, all the objectives are optimized simultaneously and the new individuals in the populations are selected using a quality indicator, as the hypervolume. IBEA [19] and HypE [1] are examples of algorithms which are based on this method.

Usually, if $M > 3$, the problem (1.1) becomes a MaOP [3]. The increase in the number of objectives is accompanied by the exponential increase in the number of non-dominated solutions, incomparable by the criteria of optimality. Hence, resulting, the convergence of the population becomes an extremely difficult task. Another problem is the generation and the selection of new individuals in the population, since the high number of non-dominated points in the population causes the selection to be random due to the lack of similarity parameters or differences between the points. In addition to the loss of selective pressure, the set of points required to represent or approximate the Pareto Front becomes very large. This increase in the number of points capable of representing the real Pareto Front implies in the increase of the population size used in the evolutionary algorithms, which becomes

unnecessarily large. Finally, visualization of the final solution in the objective space is very limited in MaOPs.

1.2.2 Introducing Preferences in MOEA

The incorporation of preferences by the DM in MOEA can be determined in a threefold manner: before the search (*a priori* approach), during the search (interactive approach) or after the search (*a posteriori* approach).

Without any prior preference provided by the DM, MOEA methods are designed to obtain an unbiased, well-distributed approximation of the entire Pareto front, a task that becomes definitely harder in MaOPs. Preference-based MOEA are designed to converge to a subset of Pareto-optimal solutions located at a specific region of the Pareto Front preferred by the DM, usually called Region of Interest (ROI), which can be defined in several ways.

For dominance-based MOEA, a popular method are MOEA based on an Achievement Scalarizing Function (ASF) [13]. Those MOEA use a reference (or goal) point in the Objective Space, representing the DM preference. Combining the information of dominance and the reference point, the MOP is transformed into a single objective problem by the minimization of the scalarizing function. In a decomposition-based MOEA, the preferences of the DM can be articulated through weight vectors[4].

1.3 Methodology

1.3.1 The Preference Cone

The proposed method uses a cone of vectors to define the ROI. A cone is defined as a geometric shape formed by a set of half-lines connecting to a common coordinate point (apex). The base is a defined plane which does not contain the apex point (coordinates). A preference cone is defined by a preferential direction vector \mathbf{v} , which corresponds to the axis of the cone. The origin or utopian point serves as the apex, and the angle τ between \mathbf{v} and the generating lines (generatrix). These elements are illustrated in Fig. 1.1.

The preferential direction vector \mathbf{v} indicates the preference of the DM. The coordinates of this vector can be the desired value for each objective or present the relative importance between each of them. For example, in a problem with three objectives, if the first objective has double importance value in comparison of the remaining two, this information translates into the vector $\mathbf{v} = (2, 1, 1)$. The aperture angle τ indicates the extension of the ROI: a small value produces a small region, providing a localized solution search. Increasing the value of this angle extends the search to a larger region. Important methods to obtain these parameters are avail-

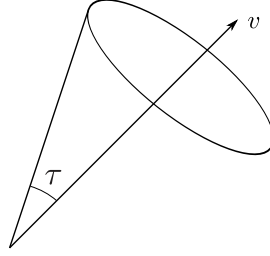


Fig. 1.1 The axis v and angle τ of the cone.

able in the literature such as the Analytical Hierarchy Process (AHP) [14] and the Stepwise Weight Assessment Ratio Analysis (SWARA) [9] methods.

Similar to the weight vectors used in Decomposition/Aggregation-based algorithms, the weight vectors of the cone are located in the hypercube $[0, 1]^M \subset \mathbb{R}^M$. The generation of weight vectors inside the preference cone is based on a steady-state evolutionary algorithm. The basic idea is to evolve an initial population W containing n vectors $\mathbf{w}_1, \dots, \mathbf{w}_n$ in the hypercube $[0, 1]^M \subset \mathbb{R}^M$ at random. Next, normalize these vectors and calculate the distance matrix $d_{i,j}$ between every pair \mathbf{w}_i and \mathbf{w}_j . For each vector $\mathbf{w}_i \in W$, create a new vector \mathbf{w}'_i from \mathbf{w}_i and calculate the distance d'_j between \mathbf{w}'_i and $\mathbf{w}_j \in W$. The new vector \mathbf{w}'_i is created by adding a Gaussian perturbation to \mathbf{w}_i . After that, remove one vector from W in order to maximize the shortest distance between the new vector \mathbf{w}'_i and the remaining vectors $\mathbf{w}_j \in W$, following an $ES(\mu + 1, \mu)$ selection scheme. The sum of the distances to the closest neighbors in W is the fitness function that guides the evolution of the set of weight vectors. Algorithm 1 presents the summarized structure of the proposed method. The details of the method are described in the following steps.

```

iter ← 0;
W = {w1, ..., wn} ← Initialize population;
W ← Normalize population(W);
di,j ← Evaluate distance between wi and wj;
φ(wi) ← Evaluate fitness function for wi ∈ W;
while stop criterion is not met do
    Choose i from {1, ..., n} at random;
    w'i ← wi + δ;
    d'j ← Evaluate distance between w'i and wj ∈ W;
    ξi ← Evaluate the angle between the new element w'i and the axis v of the cone;
    φ(w'i) ← Evaluate fitness function of the new element w'i;
    Replace the worst wj from W by w'i;
    di,j ← Update the distance matrix;
    iter ← iter + 1;
end

```

Algorithm 1: Weight Vector Generation pseudo code

Initialization: In this step the initial parameters are defined.

1. Define the number of weight vectors n to be generated.
2. Define the axis \mathbf{v} and the angle τ of the cone.
3. Define the p -norm to be used. Let $\mathbf{x} = (x_1, \dots, x_M)$ be a vector in the M -dimensional vectorial space, its p -norm is given by

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^M |x_i|^p \right)^{1/p} \quad (1.2)$$

If $p = 1$, the Manhattan norm is defined, and if $p = 2$ the Euclidean norm is described. The following equation is characterized:

$$\mathbf{y} = \frac{\mathbf{x}}{\|\mathbf{x}\|_p} \quad (1.3)$$

we define $\|\mathbf{y}\|_p = 1$. By this, $\|\mathbf{y}\|_2$ is a point on a sphere centered at $O = (0, \dots, 0)$ and unitary radius, while $\|\mathbf{y}\|_1$ is on the plane $x_1 + \dots + x_M = 1$.

4. Define the maximum number of iterations $itermax$ and the number of neighbors T . The value of T is used in the fitness function computation. After conducting some test, we ascertained the value $T = 2$.

After establishing these initial parameters, generate the initial population W at random and normalized according to (1.3). Finally, calculate the Euclidean distance matrix $d_{i,j}$ between pairs of \mathbf{w}_i and \mathbf{w}_j .

Evolutionary cycle: While the stop criteria is false, repeat the following steps below:

1. Choose an arbitrary element $\mathbf{w}_i \in W$ at random.
2. Create a new weight vector

$$\mathbf{w}'_i = \mathbf{w}_i + \delta_i \quad (1.4)$$

The perturbation vector δ_i is obtained as follows:

- a. Determine the smallest distance $d_{i,\min}$ between \mathbf{w}_i and other vectors $\mathbf{w}_j \in W$, $i \neq j$.
- b. Compute the penalty $\alpha = (1-t)a + tb$ with $t = \frac{k}{itermax}$, where k is the current iteration and $itermax$ is the maximum number of iterations. In our tests we use $a = 1.5$ and $b = 0.1$.
- c. Calculate $\delta_i = (\delta_{i1}, \dots, \delta_{iM})$, where δ_{ij} is a random variable with normal distribution of zero mean and standard deviation $\sigma_i = \alpha \times d_{i,\min}$, that is, $\delta_{ij} \sim N(0, \sigma_i)$. This choice allows the adaptation of the mutation size according to the neighborhood of the point. Moreover, it favors exploration in the beginning and local search towards the end.
- d. Normalize the new vector $\mathbf{w}'_i = (w'_{i1}, \dots, w'_{iM})$ using Equation (1.3).
3. Calculate the Euclidean distance between \mathbf{w}'_i and the remaining vectors $\mathbf{w}_j \in W$.
4. Calculate the angle ξ_i between \mathbf{w}'_i and the axis of the cone \mathbf{v} .
5. Calculate the fitness function $\phi(\mathbf{w}_j)$, $\mathbf{w}_j \in W$.

6. Let

$$\mathbf{w}_{min} = \arg \min_j \phi(\mathbf{w}_j), \mathbf{w}_j \in W \quad (1.5)$$

If $\phi(\mathbf{w}'_i) > \phi(\mathbf{w}_{min})$ replace \mathbf{w}_{min} by \mathbf{w}'_i in W and update the distance matrix $d_{i,j}$. Otherwise discard \mathbf{w}'_i .

7. Update the iteration counter.

The fitness function $\phi_i(\mathbf{w}_i)$ of $\mathbf{w}_i \in W$ is given by the sum $S_T(\mathbf{w}_i)$ of the distances from \mathbf{w}_i to its T closest neighbors in W , penalized by the angle ξ_i between \mathbf{w}'_i and the axis of the cone \mathbf{v} . If $\xi_i > \tau$, thus the vector lies outside the cone and therefore its fitness function value should be penalized. The fitness function is defined as follows:

$$\phi(\mathbf{w}_i) = S_T(\mathbf{w}_i) - M \times \max(\xi_i - \tau, 0) \quad (1.6)$$

1.4 Results and discussion

This section presents some experiments using the problems (case studies) DTLZ1, DTLZ2 [7] and WFG1 [8] with 3, 5 and 10 objectives. The aim of these experiments is to compare Dominance-based and Decomposition-Based MOEA in the exploration of a ROI in the Objective Space, using multi-objective problems. The Dominance-based algorithm selected is the g-NSGA-II [12] and the MOEA/D [17] representing the decomposition-based algorithm. For the decomposition-based algorithm, a cone of weight vectors is used instead the usual weight vector generation. As mentioned in [6], the set of weight vectors can represent the preferences of the DM for the location of the solutions in the objective space.

1.4.1 Experimental setup

The experiments were performed using the PLATEMO platform [15]. In this work, the common method of weight vector generations was substituted by the proposed novel method of generating a cone of weight vectors in the MOEA/D algorithm. All MOEA/D methods employ the same following parameters:

- Population size: The population size is $300 + 15 \times M$ individuals, where M is the number of objectives.
- Maximum number of iterations: 500 iterations;
- Genetic operators: SBX recombination ($\mu_c = 20$) and polynomial mutation ($\mu_m = 20$);

In the Decomposition Algorithm, other than the cone of vectors that define the preferences of the DM, other auxiliary weight vector cones were created. This was done by using the vectors of the canonical basis of the decision space as axis. Each extra cone consists of 15 vectors, restricted to the first orthant, i.e., for each

weight vector in the auxiliary cones $\mathbf{w} = (w_1, \dots, w_M)$, we define $w_i \geq 0 \quad i = 1, \dots, M$. These extra cones are required to guide the population to the correct location indicated by the cone of preferences. Experiments were performed with and without the extra cones and best results were obtained with the use of the auxiliary cones.

Table 1.1 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the $v1$ direction.

	DTLZ1	DTLZ2	WFG1
3 objectives			
IGD	gNSGAI 2.217e+00(3.570e+00)	3.759e-01(1.237e-03)	1.272e+00(3.184e-02)
	MOEA/D 1.354e-01(6.500e-05)	3.681e-01(2.238e-04)	1.300e+00(2.943e-03)
HV	gNSGAI 2.536e-02(3.127e-02)	2.775e-01(2.255e-03)	3.058e+01(1.859e+00)
	MOEA/D 7.683e-02(8.592e-05)	2.896e-01(2.485e-04)	3.942e+01(6.912e-02)
5 objectives			
IGD	gNSGAI 5.948e+02(1.444e+02)	1.626e+00(6.276e-01)	2.938e+00(8.431e-01)
	MOEA/D 1.480e-01(7.993e-05)	4.828e-01(5.274e-04)	1.621e+00(4.876e-02)
HV	gNSGAI 0(0)	2.832e-04(7.921e-04)	1.333e+03(1.229e+03)
	MOEA/D 2.800e-02(9.229e-05)	3.546e-01(1.938e-03)	4.881e+03(1.764e+01)
10 objectives			
IGD	gNSGAI ★	7.913e+00(2.663e+00)	6.514e+00(1.777e+00)
	MOEA/D 1.828e-01(2.365e-03)	7.574e-01(1.942e-02)	2.708e+00(1.358e-01)
HV	gNSGAI ★	0(0)	4.142e+08(8.492e+08)
	MOEA/D 1.313e-03(5.117e-05)	2.117e-01(3.408e-02)	7.096e+09(8.364e+08)

Table 1.2 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the $v2$ direction.

	DTLZ1	DTLZ2	WFG1
3 objectives			
IGD	gNSGAI 1.688e+01(3.883e+01)	4.154e-01(1.641e-03)	1.662e+00(2.155e-02)
	MOEA/D 1.457e-01(6.822e-05)	4.084e-01(1.331e-03)	1.664e+00(3.792e-03)
HV	gNSGAI 1.598e-02(2.702e-02)	2.784e-01(2.192e-03)	3.649e+01(1.076e+00)
	MOEA/D 7.676e-02(4.079e-05)	2.882e-01(1.566e-03)	3.954e+01(6.408e-02)
5 objectives			
IGD	gNSGAI 5.840e+02(1.593e+02)	1.825e+00(9.919e-01)	3.287e+00(1.025e+00)
	MOEA/D 1.524e-01(1.473e-04)	5.031e-01(3.834e-04)	1.929e+00(2.632e-02)
HV	gNSGAI 0(0)	1.922e-05(5.783e-05)	1.254e+03(1.287e+03)
	MOEA/D 2.812e-02(9.667e-05)	3.679e-01(1.667e-03)	4.855e+03(8.508e+01)
10 objectives			
IGD	gNSGAI ★	7.724e+00(2.670e+00)	7.002e+00(1.568e+00)
	MOEA/D 1.843e-01(2.685e-03)	6.848e-01(3.813e-03)	2.782e+00(6.948e-02)
HV	gNSGAI ★	0(0)	1.523e+08(5.646e+08)
	MOEA/D 1.315e-03(5.233e-05)	4.215e-01(1.012e-02)	7.435e+09(7.131e+08)

Table 1.3 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the v3 direction.

		DTLZ1	DTLZ2	WFG1
3 objectives				
IGD	gNSGAI	4.190e-01(7.944e-01)	5.309e-01(1.846e-03)	9.045e-01(1.002e-02)
	MOEA/D	1.820e-01(9.186e-05)	5.251e-01(1.507e-03)	9.602e-01(8.225e-02)
HV	gNSGAI	5.380e-02(2.732e-02)	2.782e-01(9.709e-04)	4.093e+01(3.744e-01)
	MOEA/D	7.150e-02(6.373e-05)	2.824e-01(1.500e-03)	4.002e+01(1.875e+00)
5 objectives				
IGD	gNSGAI	5.052e+02(1.477e+02)	1.788e+00(8.907e-01)	4.124e+00(1.795e+00)
	MOEA/D	1.652e-01(1.527e-04)	5.590e-01(5.380e-04)	1.608e+00(4.128e-02)
HV	gNSGAI	0(0)	2.732e-05(1.187e-04)	1.028e+03(1.057e+03)
	MOEA/D	2.687e-02(8.523e-05)	3.689e-01(1.858e-03)	4.801e+03(1.573e+02)
10 objectives				
IGD	gNSGAI	★	9.493e+00(2.404e+00)	8.698e+00(8.120e-01)
	MOEA/D	1.736e-01(3.873e-04)	7.037e-01(2.822e-03)	2.706e+00(1.346e-01)
HV	gNSGAI	★	0(0)	2.028e+07(3.188e+07)
	MOEA/D	1.549e-03(1.956e-05)	4.315e-01(1.074e-02)	6.605e+09(1.106e+09)

Table 1.4 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the v4 direction.

		DTLZ1	DTLZ2	WFG1
3 objectives				
IGD	gNSGAI	7.918e+00(1.558e+01)	4.331e-01(2.067e-03)	1.641e+00(2.516e-02)
	MOEA/D	1.502e-01(8.893e-05)	4.249e-01(1.821e-04)	1.644e+00(4.087e-03)
HV	gNSGAI	1.759e-02(2.640e-02)	2.637e-01(1.860e-03)	3.256e+01(1.655e+00)
	MOEA/D	7.637e-02(6.880e-05)	2.750e-01(3.371e-04)	3.722e+01(1.164e-01)
5 objectives				
IGD	gNSGAI	6.379e+02(1.810e+02)	2.592e+00(1.695e+00)	2.714e+00(6.954e-01)
	MOEA/D	1.576e-01(1.705e-04)	5.292e-01(8.470e-04)	1.860e+00(4.701e-02)
HV	gNSGAI	0(0)	0(0)	1.724e+03(1.162e+03)
	MOEA/D	2.818e-02(1.272e-04)	3.626e-01(1.533e-03)	4.675e+03(3.658e+01)
10 objectives				
IGD	gNSGAI	8.886e+02(6.025e+01)	8.543e+00(2.488e+00)	4.022e+00(8.363e-01)
	MOEA/D	1.737e-01(7.081e-04)	7.022e-01(2.897e-03)	2.740e+00(9.967e-02)
HV	gNSGAI	0(0)	0(0)	1.452e+09(1.820e+09)
	MOEA/D	1.586e-03(2.086e-05)	4.298e-01(1.136e-02)	7.150e+09(7.259e+08)

To analyze the performance of the algorithms, the obtained solutions are classified in three groups according to their convergence in the ROI defined by the preference cone. The first group consists of solutions located in the region defined by the preference cone, ie, the angle θ between the solution p and the axis \mathbf{v} of the cone is less than or equal to the angle τ that define the cone. The second group is composed of solutions located in the neighborhood of the region defined by the

cone. In the experiments performed, a obtained solution is in the group 2 if the angle θ between the obtained solution p and the axis \mathbf{v} of the cone is greater than τ and smaller than 2τ , i.e., $\tau < \theta < 2\tau$. All other solutions are classified in group 3. The classification of solutions into groups aims to verify the ability of each method to obtain solutions that adequately reflect the aspirations of the decision maker. Convergence and distribution of the solutions obtained will be verified through the usual metrics, restricted to solutions obtained in the selected groups.

For the experiments carried out, we considered only the solutions of group 1 and 2. All solutions p belonging to group 2 are penalized by the factor:

$$r = (\theta - \tau)^2 + e^{(1+\theta-\tau)^2} \quad (1.7)$$

where τ is the angle that defines the preference cone and θ is the angle between the obtained solution p and the axis \mathbf{v} of the preference cone. If p is a solution of group 2, it will be evaluated as $r \cdot p$. The solutions in group 1 and the penalized solutions in group 2 are analyzed using the performance metrics *Inverted Generational Distance* (IGD) [22] and *Hypervolume Indicator* (HV) [21].

1.4.2 ROI definition

For the decomposition algorithms defines the ROI is calculated by an axis \mathbf{v} and an aperture angle τ . Table 1.5 shows four different directions used as axis of the preference cone.

Table 1.5 Directions used in the experiments

Objectives	\mathbf{v}_1	\mathbf{v}_2	\mathbf{v}_3	\mathbf{v}_4
3	(1, 1, 1)	(1, 1, 0.5)	(0.1, 1, 1)	(2, 1, 1)
5	(1, 1, 1, 1, 1)	(1, 1, 1, 1, 0.5)	(0.1, 1, 1, 1, 1)	(2, 1, 1, 1, 1)
10	(1, 1, ..., 1)	(1, ..., 1, 0.5)	(0.1, 1, ..., 1)	(2, 1, ..., 1)

For all cones, including auxiliary cones, the angle τ is defined as $\tau = \arccos(1/\sqrt{M})/5$, where M is the number of objectives. The chosen directions have the following characteristics:

Direction \mathbf{v}_1 : The ROI defined in the direction \mathbf{v}_1 uses the hyperdiagonal of the Objective Space. This direction choice seeks to maximize the balance among objectives.

Direction \mathbf{v}_2 : In direction \mathbf{v}_2 , the last objective is equal to 0.5 and all the remaining are equal to 1. By choosing these values, the defined ROI will be further away from the last objective, but is balanced in relation to the others.

Direction \mathbf{v}_3 : The direction \mathbf{v}_3 presents a low value in the first coordinate (only 1/10 of the value of the other objectives). As a result, the ROI defined in this direction will contain few points (or no point) in this region.

Direction \mathbf{v}_4 : In the direction \mathbf{v}_4 , the first coordinate is equal to 2 and all the others are equal to 1. By this, the defined ROI will be closer to the first objective and farther from the remaining objectives.

For the g-NSGA-II algorithm, the ROI is defined by a reference point \mathbf{G} . Since all problem have the true pareto front well defined, the \mathbf{G} point is easy to define: let $\mathcal{G} = \{\mathbf{g}^1 \dots \mathbf{g}^q\}$ be points in the PF located in the ROI defined by the preference cone and set \mathbf{G} the ideal point of \mathcal{G} .

Thirty instances of each algorithm were performed for each ROI. Table 1.5 presents the direction vectors that define the ROI used in the experiments and Table 1.6 presents the success rate of each experiment. An experiment was considered successful if at least one solution of group 1 or 2 was found. Tables 1.1 to 1.4 present the average and the standard deviation of the IGD and HV metrics of the tests performed, highlighting the best and the worst result. A \star marker is used when no results from group 1 or 2 were found.

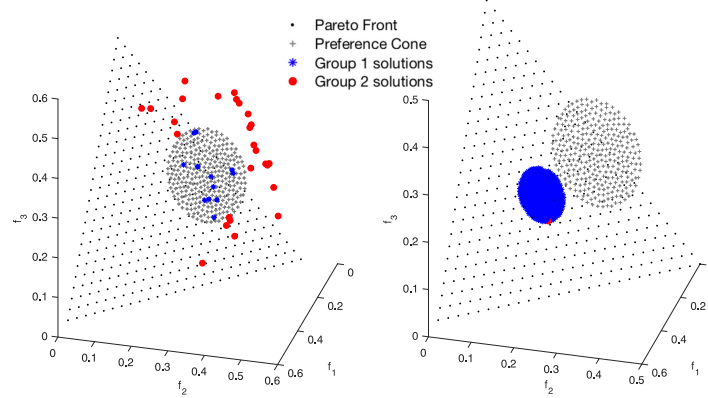
Table 1.6 Success rate for the problems with 3, 5 and 10 objectives.

		3 objectives			5 objectives			10 objectives		
		DTLZ1	DTLZ2	WFG1	DTLZ1	DTLZ2	WFG1	DTLZ1	DTLZ2	WFG1
v1	gNSGAII	0.60	1.00	1.00	0.57	1.00	1.00	0.00	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
v2	gNSGAII	1.00	1.00	1.00	0.90	1.00	1.00	0.00	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
v3	gNSGAII	0.87	1.00	1.00	0.70	1.00	1.00	0.00	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
v4	gNSGAII	0.73	1.00	1.00	0.93	1.00	1.00	0.07	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

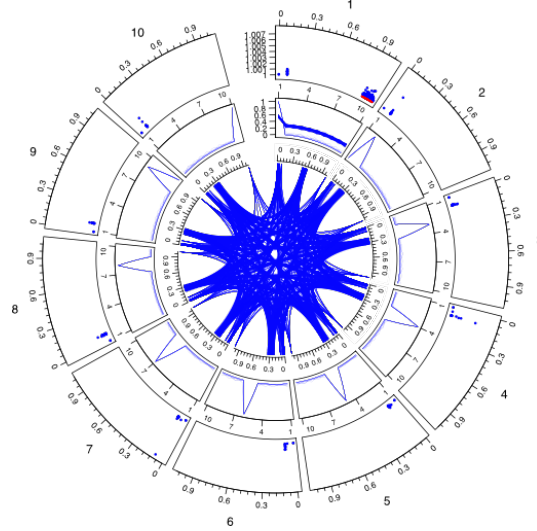
Figure 1.3 presents the obtained solution (blue dots) for the DTLZ2 problem with 10 objectives in \mathbf{v}_4 direction and the ROI (red dots) using CAP-vis visualization tool [11]. The points represented at the beginning of track C of each sector represent the solutions found in the auxiliary cones. The ROI is entirely localized in sector 1 and it is possible to observe the alignment of the solutions obtained with this region.

1.4.3 Discussion

The decomposition algorithm proposed in this work, using the cone of weight vectors obtained better results than the dominance algorithm using ASF strategy. Figure 1.2

Fig. 1.2 Obtained solutions (blue dots) for the DTLZ1 problem with 3 objectives in \mathbf{v}_1 direction.

depicts the obtained solutions of a single run for each method in the \mathbf{v}_1 direction, highlighting group 1 and group 2 solutions for the DTLZ1 problem. Algorithm gNSGAII presented better performance only in the three objective WFG1 problem, regardless of the ROI analyzed. However, in other problems, this method presented a high variance of results in most cases. This phenomenon is caused by the prevalence of solutions of Group 2 which due to its penalty makes the value of its metric increased. Moreover, the results obtained reinforce the inadequacy of the algorithms based on dominance for problems with many objectives.

Fig. 1.3 Obtained solutions for the DTLZ2 problem with 10 objectives in \mathbf{v}_4 direction

From the experiments conducted, it can be concluded that different regions of the objective space present different challenges for the optimizer of the same problem. As an example, the value of the IGD metric obtained in the region defined by the direction v_3 in the problem WFG1 indicates that in this region the algorithms used can obtain solutions with better convergence than others directions that were analyzed.

1.5 Conclusions

Due to the fundamental characteristics of many-objective optimization problems, obtaining a well-distributed and representative approximation of the Pareto Front is a hard task. Moreover, the analysis of the solutions obtained and the subsequent choice of a particular solution are challenging. Moreover, defining exactly the preferences of the DM in an *a priori* approach is difficult in most practical cases. Researching a noncommittal approach becomes attractive in such a scenario, in which the search for solutions in a specific region of the objective space that corresponds to the aspirations of the DM is a way to make this type of optimization problem less difficult.

This paper presented a new preference-based methodology to perform the search for solutions in the ROI, defined by a preference cone in the Objective Space. The exploration of the ROI using the preference cone presented good convergence and dispersion of the solutions, showing that this is an adequate methodology for preference-guided many-objective optimization. In addition, this approach of determining the ROI is more intuitive and is able to reflect the preferences of the decision-maker.

References

1. Bader, J., Zitzler, E.: HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation* **19**(1), 45–76 (2011). DOI 10.1162/evco_a_00009
2. Batista, L.S., Campelo, F., Guimarães, F.G., Ramírez, J.A.: A comparison of dominance criteria in many-objective optimization problems. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 2359–2366 (2011). DOI 10.1109/CEC.2011.5949909
3. Bechikh, S., Kessentini, M., Said, L.B., Ghédira, K.: Preference incorporation in evolutionary multiobjective optimization. In: A.R. Hurson (ed.) *Advances in Computers, Advances in Computers*, vol. 98, pp. 141–207. Elsevier (2015). DOI 10.1016/bs.adcom.2015.03.001
4. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* **20**(5), 773–791 (2016). DOI 10.1109/tevc.2016.2519378
5. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA (2001)
6. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* **18**(4), 577–601 (2014)

7. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: A. Abraham, L. Jain, R. Goldberg (eds.) *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pp. 105–145. Springer London, London (2005). DOI 10.1007/1-84628-137-7_6
8. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* **10**(5), 477–506 (2006). DOI 10.1109/tevc.2005.861417
9. Keršulienė, V., Zavadskas, E.K., Turskis, Z.: Selection of rational dispute resolution method by applying new step-wise weight assessment ratio analysis (swara). *Journal of Business Economics and Management* **11**(2), 243–258 (2010). DOI 10.3846/jbem.2010.12
10. Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys* **48**(1), 1–35 (2015). DOI 10.1145/2792984
11. Meneghini, I.R., Koochaksaraei, R.H., Guimarães, F.G., Gaspar-Cunha, A.: Information to the eye of the beholder: Data visualization for many-objective optimization. In: 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE (2018). DOI 10.1109/cec.2018.8477889
12. Molina, J., Santana, L.V., Hernández-Díaz, A.G., Coello, C.A.C., Caballero, R.: g-dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research* **197**(2), 685–692 (2009). DOI 10.1016/j.ejor.2008.07.015
13. Ruiz, A.B., Saborido, R., Luque, M.: A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. *Journal of Global Optimization* **62**(1), 101–129 (2014). DOI 10.1007/s10898-014-0214-y
14. Saaty, T.L.: *The analytic hierarchy process : planning, priority setting, resource allocation*. McGraw-Hill International Book Co., New York; London (1980)
15. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine* **12**(4), 73–87 (2017). DOI 10.1109/mci.2017.2742868
16. Trivedi, A., Srinivasan, D., Sanyal, K., Ghosh, A.: A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation* **21**(3), 440–462 (2017). DOI 10.1109/tevc.2016.2608507
17. Zhang, Q., Li, H.: MOEA/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* **11**(6), 712–731 (2007). DOI 10.1109/tevc.2007.892759
18. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* **1**(1), 32–49 (2011). DOI 10.1016/j.swevo.2011.03.001
19. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pp. 832–842. Springer (2004)
20. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: K. Giannakoglou, et al. (eds.) *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pp. 95–100. International Center for Numerical Methods in Engineering (CIMNE) (2002)
21. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study. In: A.E. Eiben, T. Bäck, M. Schoenauer, H.P. Schwefel (eds.) *Lecture Notes in Computer Science*, pp. 292–301. Springer Berlin Heidelberg, Berlin, Heidelberg (1998). DOI 10.1007/bfb0056872
22. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* **7**(2), 117–132 (2003). DOI 10.1109/tevc.2003.810758